Complexité avancée - TD 3

Benjamin Bordais

October 06, 2021

We recall the space-hierarchy theorem.

**Theorem 1** (Space-hierarchy theorem). For two space-constructible functions \( f \) and \( g \) such that \( f = o(g) \), we have \( \text{DSPACE}(f) \subsetneq \text{DSPACE}(g) \).

**Exercise 1** (Poly-logarithmic space).

1. Let \( \text{polyL} = \bigcup_{k \in \mathbb{N}} \text{SPACE} \left( \log^k \right) \). Show that \( \text{polyL} \) does not have a complete problem for logarithmic space reduction.

2. Recall that \( \text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{SPACE} \left( n^k \right) \). Does \( \text{PSPACE} \) have a complete problem for logarithmic space reduction? Why doesn’t the proof of the previous question apply to \( \text{PSPACE} \)?

**Exercise 2** (Padding argument).

1. Show that if \( \text{DSPACE}(n^c) \subseteq \text{NP} \) for some \( c > 0 \), then \( \text{PSPACE} \subseteq \text{NP} \).

Hint: for \( L \in \text{DSPACE}(n^k) \) one may consider the language \( \tilde{L} = \{(x, w_x) \mid x \in L\} \). where \( w_x \) is a word written in unary.

2. Deduce that \( \text{DSPACE}(n^c) \neq \text{NP} \).

**Exercise 3** (On the existence of One-way function). A one-way function is a bijection \( f \) from \( k \)-bit integers to \( k \)-bit integers such that \( f \) is computable in polynomial time, but \( f^{-1} \) is not. Prove that for all one-way functions \( f \), we have

\[
A := \{(x, y) \mid f^{-1}(x) < y \} \in (\text{NP} \cap \text{coNP}) \setminus \text{P}
\]

**Exercise 4** (Regular languages). Let \( \text{REG} \) denote the set regular/rational languages.

1. Show that for all \( L \in \text{REG} \), \( L \) is recognized by a TM running in space 0 and time \( n + 1 \).

2. Exhibit a language recognized by a TM running in space \( \log n \) and time \( O(n) \) that is not in \( \text{REG} \).

**Exercise 5** (Yet another \( \text{NL} \)-complete problem). For a finite set \( X \), a subset \( S \subseteq X \), and a binary operator \( * : X \times X \rightarrow X \) defined on \( X \), we inductively define \( S_{0,*} := S \) and \( S_{i+1,*} := S_{i,*} \cup \{x * y \mid x, y \in S_{i,*}\} \). The closure of \( S \) with regard to \( * \) is the set \( S_* = \bigcup_{i \in \mathbb{N}} S_{i,*} \).

Show that the following problem is \( \text{NL} \)-complete.

- **Input:** A finite set \( X \), a binary operation \( * : X \times X \rightarrow X \) that is associative (i.e. \( (x * y) * z = x * (y * z) \) for all \( x, y, z \in X \)), a subset \( S \subseteq X \) and a target \( t \in X \).

\(^1\)Note that, from this, we can deduce that \( \text{polyL} \neq \text{P} \).

\(^2\)In fact, regular languages exactly correspond to languages that can be recognized in such a way.
• Output: Yes if and only if \( t \in S^\ast \).

Exercise 6 (Solving reachability games). A two player (turn-based) game is a directed graph \( G = (V,E) \) where the set of vertices \( V = V_A \uplus V_B \) is partitioned into vertices belonging to Player A (i.e. \( V_A \)) and vertices belonging to Player B (i.e. \( V_B \)) with a distinguished vertex \( v_0 \in V \) that is the starting vertex. The graph is non-blocking in the sense that every vertex has a successor, i.e. \( \text{Succ}(v) = \{ v' \in V \mid (v,v') \in E \} \neq \emptyset \) for all \( v \in V \). A play then corresponds to a finite or infinite path \( \rho = v_0 \cdot v_1 \cdots \in V^\ast \cup V^\omega \) with \( v_0 \) is the starting vertex. If the play is at a vertex \( v_i \in V_A \) then it is Player A’s turn to choose the next vertex \( v_{i+1} \in \text{Succ}(v_i) \), while it is Player B’s turn if \( v_i \in V_B \). A winning condition determines when a play is winning for Player A (we consider win/lose games, hence if Player A does not win, Player B does). A Player \( C \in \{ A,B \} \) has a winning strategy (or wins) from a vertex \( v \in V \) if she can choose the next move in all vertices in \( V_C \) such that she wins in any play that starts in \( v \).

1. Assume that the winning condition is a reachability objective: given a target set of states \( T \subseteq V \), Player A wins if and only if a state in \( T \) is seen at some point. Show that deciding the winner of a reachability game from the vertex \( v_0 \in V \) can be done in polynomial time.
   
   Hint: construct inductively the set of vertices from which Player A can ensure to get closer to the target \( T \) (that is called the attractor of the set \( T \)).

2. Consider some \( k \in \mathbb{N} \). A \( k \)-generalized reachability condition is the following: given \( k \) target sets of states \( T_1,\ldots,T_k \subseteq V \), Player A wins if and only if, for all \( 1 \leq i \leq k \), a state in \( T_i \) is seen at some point. Show that deciding the winner of a \( k \)-generalized reachability game from the vertex \( v_0 \in V \) can be done in polynomial time.

Due October 13, 2021

Homework (Solving generalized reachability games). In the two-player turn-based setting:

1. A generalized reachability condition is the following: given several target sets of states \( T_1,\ldots,T_k \subseteq V \), Player A wins if and only if, for all \( 1 \leq i \leq k \), a state in \( T_i \) is seen at some point. The difference with the \( k \)-generalized reachability objective is that the number of targets is not bounded a priori. Show that deciding the winner of a generalized reachability game is \( \text{PSPACE} \)-complete.
   
   Hint: for the \( \text{PSPACE} \) membership, you can use without a proof that if Player A wins with \( k \) target sets of states, she can win in at most \( k \cdot n \) steps where \( n := |V| \). For the hardness, you may reduce from \( \text{TQBF} \) where the formula is in \( \text{CNF} \).

2. Show that deciding the winner of a generalized reachability game when \( V_B = \emptyset \) (i.e. only Player A is playing) is \( \text{NP} \)-hard.